

**Amendment to the Claims**

Please amend Claims 12, 20, 30, 36-38 as follows:

- 5 1. (previously presented) A method of managing a database that includes a plurality of sections, each of the sections comprising a plurality of data records, the method comprising:
- receiving a new data record and a key that is associated with the new data record;
- 10 identifying one of the sections based upon the associated key of the new data record;
- determining if said new data record fits in an unused space on said identified section;
- if said new data record fits in said unused space, then storing said new data
- 15 record in said identified section;
- if a size of said new data record is greater than a size of said unused space, then ranking all data records on said identified section according to a ranking function;
- summing sizes of said all data records below rank of said new data record;
- if said sum is not greater than said size of said new data record, then ending
- 20 process; and
- if said sum is greater than said size of said new data record, then deleting one or more data records from the identified section and storing the new data record in the identified section.

2. (canceled)

3. (previously presented) The method of Claim 1, wherein the ranking function is a least recently used algorithm.

5

4. (previously presented) The method of Claim 1, wherein the ranking function is a function of the statistical properties of the data being stored.

5. (original) The method of Claim 1, wherein each of the plurality of sections is an integer multiple of the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

6. (original) The method of Claim 1, wherein each of the sections is about the same page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

7. (original) The method of Claim 1, additionally comprising allocating a contiguous memory space to contain each of the sections.

8. (previously presented) A program storage device storing program instructions that when executed perform the program comprising:

receiving a new data record and a key that is associated with the new data record;

identifying a section from a plurality of sections, the identifying based upon the associated key of the new data record;

determining if said new data record fits in an unused space on said identified section;

5 if said new data record fits in said unused space, then storing said new data record in said identified section;

if a size of said new data record is greater than a size of said unused space, then ranking all data records on said identified section according to a ranking function;

summing sizes of said all data records below rank of said new data record;

10 if said sum is not greater than said size of said new data record, then ending process; and

if said sum is greater than said size of said new data record, then deleting one or more data records from the identified section and storing the new data record in the identified section.

15

9. (canceled)

10. (previously presented) The program storage device of Claim 8, wherein the ranking scheme identifies which ones of the data records are the least recently used.

20

11. (original) The method of Claim 8, wherein each of the sections is about the same size that is used by an operating system to transfer data between a primary storage and a secondary storage.

12. (currently amended) A database system for managing data records, the system comprising:

a plurality of sections, each of the sections being about the same memory size that is used by an operating system to transfer data between a primary storage and a  
5 secondary storage; and

a control program which receives a request for the storage of a data record, the control program selecting one of the sections based upon a key and storing the data record in the selected section;

wherein the control program:

10 determines if said data record fits in an unused space on said selected section;

determines if said data record fits in said unused space, then ~~storing~~ stores said data record in said selected section;

15 if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;

sums sizes of said all data records below rank of said data record;

if said sum is not greater than said size of said data record, then ends process; and

20 if said sum is greater than said size of said data record, then deletes one or more data records from the selected section and stores the data record in the selected section.

13. (canceled)

14. (previously presented) The database system of Claim 12, wherein the ranking function determines a last access time for each of the data records or the selected sections.

5 15. (original) The database system of Claim 12, wherein at least one of the sections includes at least one item of section information.

16. (original) The database system of Claim 15, wherein the section information includes the number of data records that are contained in the section.

10

17. (original) The database system of Claim 15, wherein the section information includes an offset from the beginning of the section to the first unused position within the section.

15 18. (original) The database system of Claim 15, wherein the section information includes a section number that is associated with the section.

19. (original) The database system of Claim 12, additionally comprising a client application which provides the storage request of the data record and the key to the  
20 control program.

20. (currently amended) A database system for managing information items, the system comprising:

a plurality of sections; and

a control program which receives a request for the storage of a data record, the control program selecting one of the sections and storing the data record in the selected section, the control program:

determining if said data record fits in an unused space on said selected section;

5        determining if said data record fits in said unused space, then storing said data record in said selected section;

determining if a size of said data record is greater than a size of said unused space, then ranking all data records on said selected section according to a ranking function;

10        summing sizes of said all data records below rank of said data record;

if said sum is not greater than said size of said data record, then ending process;

and

if said sum is greater than said size of said data record, then deleting one or more data records from the selected section and storing the data record in the selected  
15        section.

21.        (original) The database system of Claim 20, wherein the ranking function determines an access time for each of the data records or the selected sections.

20        22.        (original) The database system of Claim 20, wherein each of the data records stores at least one user profile.

23.        (original) The database system of Claim 20, wherein at least one of the sections includes at least one record of section information.

24. (original) The database system of Claim 23, wherein the section information includes the number of data records that are contained in the section.

5 25. (original) The database system of Claim 23, wherein the section information includes an offset from the beginning of the section to the first unused position within the section.

26. (original) The database system of Claim 23, wherein the section information  
10 includes a section number that is associated with the section.

27. (original) The database system of Claim 20, additionally comprising a client application which provides the storage request of the data record and the key to the control program.

15

28. (original) The database system of Claim 20, wherein the size of each of the sections is an integer multiple to the transfer size that is used by an operating system to transfer data between a primary storage and a secondary storage.

20 29. (original) The database system of Claim 20, wherein the size of each of the sections is about equal to the transfer size that is used by an operating system to transfer data between a primary storage and a secondary storage.

30. (currently amended) A system for managing a database that includes a plurality of sections, each of the sections comprising a plurality of data records, the ~~method~~ system comprising:

means for receiving one or more new data records, each of the new data records

5 having an associated key;

means for identifying one of the sections based upon the associated key of the new data record;

means for determining if said new data record fits in an unused space on said;

10 If said new data record fits in said unused space, then means for storing said new data record in said identified section;

if a size of said new data record is greater than a size of said unused space, then means for ranking all data records on said identified section according to a ranking function;

15 means for summing sizes of said all data records below rank of said new data record;

if said sum is not greater than said size of said new data record, then means for ending process; and

20 If said sum is greater than said size of said new data record, then means for deleting one or more data records from said identified section and means for storing said new data record in the identified section.

31. (canceled)



32. (original) The system of Claim 30, wherein the ranking function identifies which ones of the data records that are the least recently used.

33. (original) The system of Claim 30, wherein the database occupies a single  
5 contiguous physical memory space.

34. (original) The system of Claim 30, wherein the size of each of the sections is an integer multiple to the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

10

35. (original) The system of Claim 30, wherein the size of each of the sections is about equal to the page size that is used by an operating system to transfer data between a primary storage and a secondary storage.

15 36. (currently amended) A database system for managing information records, the system comprising:

a primary storage;

a secondary storage having a plurality of pages;

a plurality of sections, wherein each of the sections is adapted to contain one or  
20 more data records, and wherein each of the sections resides in the secondary storage on one of the plurality of pages; and

a control program which receives a request for the retrieval of a data record, the control program retrieving the data record from the secondary storage and storing the

data record in the primary storage, wherein the retrieval operation reads at most one page from the secondary storage;

a database data structure having a plurality of sections, each of the sections residing on one of the pages in any of the primary storage and/or the secondary

5 storage; and

a database manager which receives requests from the client application to store a data record in the database data structure, wherein the database manager selects one of the sections and stores the data record in the selected section, wherein the database manager:

10 determines if said data record fits in an unused space on said selected section;

determines if said data record fits in said unused space, then stores said data record in said selected section;

15 determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;

sums sizes of said all data records below rank of said data record;

if said sum is not greater than said size of said data record, then ends process; and

20 if said sum is greater than said size of said data record, then deletes one or more data records from the selected section and stores the data record in the selected section.

37. (currently amended) A database system for managing information items, the system comprising:

a primary storage;

a secondary storage having a plurality of pages;

5 a plurality of sections, each of the sections independent of each of the other sections such that an error in one of the sections does not affect any of the other sections; and

a control program which receives a request for the retrieval of a data record, the control program retrieving the data record from the secondary storage so that the data  
10 record is stored in the primary storage;

a database data structure having a plurality of sections, each of the sections residing on one of the pages in any of the primary storage and/or the secondary storage; and

a database manager which receives requests from the client application to store  
15 a data record in the database data structure, wherein the database manager selects one of the sections and stores the data record in the selected section, wherein the database manager:

determines if said data record fits in an unused space on said selected section;

20 determines if said data record fits in said unused space, then stores said data record in said selected section;

determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking function;

sums sizes of said all data records below rank of said data record;

if said sum is not greater than said size of said data record, then ends process; and

if said sum is greater than said size of said data record, then deletes one or  
5 more data records from the selected section and stores the data record in the selected section.

38. (currently amended) A database system for managing information items, the system comprising:

10 a client application;

a primary storage comprising a plurality of pages;

a secondary storage comprising a plurality of pages;

a caching subsystem for copying pages from the secondary storage to the pages  
in the primary storage and vice-versa;

15 a database data structure having a plurality of sections, each of the sections residing on one of the pages in any of: the primary storage and the secondary storage; and

a database manager which receives requests from the client application to store  
a data record in the database data structure, wherein the database manager selects  
20 one of the sections and stores the data record in the selected section, wherein the database manager:

determines if said data record fits in an unused space on said selected section;

determines if said data record fits in said unused space, then stores said data record in said selected section;

determines if a size of said data record is greater than a size of said unused space, then ranks all data records on said selected section according to a ranking  
5 function;

sums sizes of said all data records below rank of said data record;

If said sum is not greater than said size of said data record, then ends process; and

If said sum is greater than said size of said data record, then deletes one or  
10 more data records from the selected section and stores the data record in the selected section.